AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A system for sorting incoming data packets in real time, comprising:

assignment means, operable only on packet records containing information about said packets, for assigning an exit order to said packets in real time;

~~queue means~~ a sole queue for handling all packet records which enter said system responsive to said assignment means for storing and arranging said packet records in said exit order; and

memory means for storing said packets or data portions thereof; said packets or data portions being output from the memory means in accordance with the exit order of the corresponding packet records in the sole queue ~~means~~.

2. (Previously Presented) A system as claimed in claim 1, wherein the assignment means is responsive to information contained within a packet whereby to determine an exit order number for that packet.

3. (Previously Presented) A system as claimed in claim 1, wherein the assignment means is responsive to information contained in a table whereby to determine an exit order number for that packet.

4. (Previously Presented) A system as claimed in claim 1, wherein the assignment means is responsive to information associated with a data packet stream in

which said packet is located whereby to determine an exit order number for that packet.

5. (Currently Amended) A system as claimed in claim 1, wherein said assignment means is adapted to insert packet records in said sole queue ~~means~~ in exit order.

6. (Cancelled)

7. (Currently Amended) A system as claimed in claim [[6]]1, wherein said ~~single~~ sole queue provides a plurality of virtual queues.

8. (Currently Amended) A system as claimed in claim 1, wherein said sole queue ~~means~~ comprises a queue manager adapted to insert packet records into said sole queue ~~means~~ in exit order.

9. (Currently Amended) A system as claimed in claim 1, further comprising means to drop certain packet records before being output from said sole queue ~~means~~.

10. (Currently Amended) A system as claimed in claim 1, further comprising means to drop certain packet records before being queued in said sole queue ~~means~~.

11. (Previously Presented) A system as claimed in claim 1, wherein said assignment means comprises a processor.

12. (Previously Presented) A system as claimed in claim 11, wherein said processor comprises a parallel processor.

13. (Previously Presented) A system as claimed in claim 12, wherein said

parallel processor is an array processor comprising one or more arrays of processor elements.

14. (Previously Presented) A system as claimed in claim 13 wherein said array processor is a SIMD processor.

15. (Previously Presented) A system as claimed in claim 12, further comprising means to provide access for said parallel processor to shared state.

16. (Original) A system as claimed in claim 15, further comprising a state engine to control said access to said shared state.

17. (Previously Presented) A system as claimed in claim 11, further comprising tables of information for assigning said exit order to said packets, wherein said tables are stored locally to said processor.

18. (Previously Presented) A system as claimed in claim 17, wherein said tables are for the same class of service on said processor.

19. (Previously Presented) A system as claimed in claim 17, wherein said tables are for varying classes of service on said processor.

20. (Cancelled)

21. (Previously Presented) A system as claimed in claim 1, wherein said assignment means implements algorithms for packet scheduling in accordance with predetermined criteria, including WFQ, DFR, congestion avoidance (e.g. WRED) or other prioritisation and sorting.

22. (Currently Amended) A method for sorting incoming data packets in real time, the method comprising the steps of:

assigning incoming data packets an exit order in real time from packet records only, each of said packet records containing information about a corresponding packet;

storing and arranging in a sole queue ~~means~~, which handles all packet records entering said system, said packet records for output in said exit order;

storing in memory means said packets or data portions thereof; and

outputting packets or data portions from the memory means in accordance with the exit order of the corresponding packet records in the sole queue ~~means~~.

23. (Previously Presented) A method as claimed in claim 22, wherein the assigning is responsive to information contained within a packet whereby to assign an exit order number for that packet.

24. (Previously Presented) A method as claimed in claim 22, wherein the assigning is responsive to information contained in a table whereby to determine an exit order number for that packet.

25. (Previously Presented) A method as claimed in claim 22, wherein the assigning is responsive to information associated with a data packet stream in which said packet is located whereby to determine an exit order number for that packet.

26. (Currently Amended) A method as claimed in claim 22, wherein said packet records are inserted into [[a]] said sole queue ~~means~~ for output in exit order determined by the means performing the assigning.

27. (Currently Amended) A method as claimed in claim 22, comprising

inserting said packet records into [[a]] said sole queue ~~means~~ for output in exit order under control of a queue manager.

28. (Cancelled)

29. (Currently Amended) A method as claimed in claim [[28]]22, further comprising providing a plurality of virtual queues by means of said ~~single output~~ sole queue.

30. (Currently Amended) A method as claimed in claim 22, further comprising dropping certain packets before the corresponding packet records are output from said sole queue ~~means~~.

31. (Currently Amended) A method as claimed in claim 22, further comprising dropping certain packets before the corresponding packet records are queued in said sole queue ~~means~~.

32. (Previously Presented) A method as claimed in claim 22, wherein said assigning is performed by a processor.

33. (Previously Presented) A method as claimed in claim 32, wherein said processor is a parallel processor.

34. (Previously Presented) A method as claimed in claim 33, wherein said parallel processor is an array processor comprising one or more arrays of processor elements.

35. (Previously Presented) A method as claimed in claim 34, wherein said

array processor is a SIMD processor.

36. (Previously Presented) A method as claimed in claim 32, further comprising providing access for said processor to shared state under control of a state engine.

37. (Currently Amended) A method as claimed in claim 32, further comprising providing tables of information for assigning said packets, wherein said tables are stored locally to [[said]] processor elements of said processor.

38. (Previously Presented) A method as claimed in claim 37, wherein said tables are for the same class of service on said processor elements.

39. (Previously Presented) A method as claimed in claim 37, wherein said tables are for varying classes of service on said processor elements.

Claims 40 – 41 (Cancelled)

42. (Currently Amended) A computer system, comprising a data handling system for sorting incoming data packets in real time, the computer system comprising:

assignment means, operable only on packet records containing information about said packets, for assigning an exit order to said packets in real time;

a sole queue ~~means~~ for handling all packet records which enter said system and responsive to said assignment means for storing and arranging said packet records for output in said exit order; and

memory means for storing said packets or data portions thereof;

said packets or data portions being output from the memory means in accordance with the exit order of the corresponding packet records in the sole

queue ~~means~~.

43. (Currently Amended) A network processing system, comprising a data handling system for sorting incoming data packets in real time, the data handling system comprising;

assignment means, operable only on packet records containing information about said packets, for assigning an exit order to said packets in real time;

a sole queue ~~means~~ for handling all packet records which enter said system and responsive to said assignment means for storing and arranging said packet records for output in said exit order; and

memory means for storing said packets or data portions thereof;

said packets or data portions being output from the memory means in accordance with the exit order of the corresponding packet records in the sole queue ~~means~~.

44. (Currently Amended) A computer system adapted to perform sorting of incoming data packets in real time in which said data packets are assigned an exit order and output in said exit order, the computer system comprising:

assignment means, operable only on packet records containing information about said packets, for assigning an exit order to said packets in real time;

a sole queue ~~means~~ for handling all packet records which enter said system and responsive to said assignment means for storing and arranging said packet records for output in said exit order; and

memory means for storing said packets or data portions thereof;

said packets or data portions being output from the memory means in accordance with the exit order of the corresponding packet records in the sole queue ~~means~~.

45. (Currently Amended) A network processing system adapted to perform sorting of incoming data packets in real time in which said data packets are assigned an exit order and are output in said exit order, the network processing system comprising:

assignment means, operable only on packet records containing information about said packets, for assigning an exit order to said packets in real time;

a sole queue ~~means~~ for handling all packet records which enter said system and responsive to said assignment means for storing and arranging said packet records for output in said exit order; and

memory means for storing said packets or data portions thereof;

said packets or data portions being output from the memory means in accordance with the exit order of the corresponding packet records in the sole queue ~~means~~.

46. (Original) A computer system as claimed in claim 42 implemented as one or more silicon integrated circuits.

47. (Currently Amended) A computer-readable medium containing instructions which, when executed on a computer processor or processor element, perform sorting of incoming data packets in real time in which said data packets are assigned an exit order and are output in said exit order, and performs the steps of:

assigning incoming data packets an exit order in real time from packet records only, each of said packet records containing information about a corresponding packet;

storing and arranging in a sole queue ~~means~~, which handles all packet records which enter said system, said packet records for output in said assigned exit order;

storing in memory means said packets or data portions thereof; and outputting packets or data portions from the memory means in accordance with the exit order of the corresponding packet record in the sole queue ~~means~~.

48. (Previously Presented) A system as claimed in claim 13, further comprising tables of information for sorting said packets or said packet records, wherein said tables are stored locally to each processor element of said parallel processor.

49. (Previously Presented) A system as claimed in claim 48, wherein said tables are for the same class of service on each processor element of said parallel processor.

50. (Previously Presented) A system as claimed in claim 48, wherein said tables are for varying classes of service on different processor elements of said parallel processor.

51. (Previously Presented) A system as claimed in claim 48, wherein said processor elements share information from their respective tables, such that:

(a) the information held in the table in one processor element is accessible by other processing element(s) of said parallel processor; and

(b) processor elements have access to other processor elements in said parallel processor, whereby processor elements can perform table lookups on behalf of other processor elements of said parallel processor.

52. (Previously Presented) A method as claimed in claim 34, further comprising providing tables of information for sorting said packets or said packet records, wherein said tables are stored locally to each processor element of said parallel processor.

53. (Previously Presented) A method as claimed in claim 52, wherein said tables are for the same class of service on each processor element of said parallel processor.

54. (Previously Presented) A method as claimed in claim 52, wherein said tables are for varying classes of service on each processor element of a parallel processor.

55. (Previously Presented) A method as claimed in claim 52, wherein said processor elements share information from their respective tables, such that:

(a) the information held in the table for one processor element is made directly accessible to other processor element(s) of said parallel processor; and

(b) access is provided for said processor elements to tables in other processor elements, whereby processor elements can perform table lookups on behalf of other processor elements of said parallel processor.

56. (Currently Amended) A system for sorting and storing incoming data packets comprising:

an input port which receives record portions of said incoming data packets in an input order;

an assignment processor which operates on said record portions of said incoming data packets to assign an exit order for said incoming data packets, wherein said exit order differs from said input order; and

a single queue, responsive to said assignment processor, for storing said record portions of all of said incoming data packets which enter said system in said exit order.

57. (Previously Presented) The system of claim 56, wherein said system is implemented on a single chip.

58. (Previously Presented) The system of claim 56, further comprising:

an input device, upstream of said input port, for receiving said incoming data

packets and forwarding said record portions of said data packets to said input port; and

a memory hub, connected to said input device, for receiving and storing one of: (a) said incoming data packets or (b) data portions of said incoming data packets.

59. (Previously Presented) The system of claim 58, further comprising:

an output device, connected to said single queue and said memory hub, for outputting said data packets from said system in said exit order.

60. (Currently Amended) A method for sorting and storing incoming data packets, the method comprising:

receiving, using an input port, record portions of said incoming data packets in an input order;

operating, using an assignment processor, on said record portions of said incoming data packets to assign an exit order for said incoming data packets, wherein said exit order differs from said input order; and

storing, using a single queue responsive to said assignment processor, said record portions of all of said incoming data packets which enter said system in said exit order.

61. (Previously Presented) The method of claim 60, wherein said method is implemented using a single chip.

62. (Previously Presented) The method of claim 60, further comprising:

receiving said incoming data packets and forwarding said record portions of said data packets to said input port using an input device upstream of said input port; and

receiving and storing, using a memory hub connected to said input device, one of (a) said incoming data packets or (b) data portions of said incoming data packets.

63. (Previously Presented) The method of claim 62, further comprising:

outputting, using an output device connected to said single queue and said memory hub, said data packets from said system in said exit order.